(2)

CHEMICAL RESEARCH, DEVELOPMENT & ENGINEERING CENTER

CRDEC-TN-024

BACK-PROPAGATION NETWORK FOR ANALOG SIGNAL SEPARATION IN HIGH NOISE ENVIRONMENTS

Richard G. Vanderbeek
Alice M. Harper

RESEARCH DIRECTORATE

July 1992

U.S. ARMY
ARMAMENT
MUNITIONS
CHEMICAL COMMAND

Aberdeen Proving Ground, Maryland 21010-5423

Disclaimer

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorizing documents.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	1992 July	Final, 91 Aug - 92 Jan

4. TITLE AND SUBTITLE
Back-Propagation Network for Analog Signal Separation in High Noise Environments

5. FUNDING NUMBERS
C-DAAD05-91-P6564

6. AUTHOR(S)
Vanderbeek, Richard G., and Harper, Alice M.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
CDR, CRDEC, ATTN: SMCCR-RSP-C, APG, MD 21010-5423

8. PERFORMING ORGANIZATION REPORT NUMBER
CRDEC-TN-024

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A back-propagation network is compared with principal components regression and prefiltered linear regression to demonstrate its ability to separate overlapped analog signals in high noise environments. Specifically, the signals tested were synthetically generated chemical mixture spectra that simulate the type of data obtained from chromatography and photospectrometry. The individual spectrum are heavily overlapped and 30 percent random noise and a random dc has been added to them. The comparisons were made for data sets comprised of two, three, and four overlapping spectrum.

14. SUBJECT TERMS
Neural networks Chromatography
Back-propagation

15. NUMBER OF PAGES
13

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Blank

PREFACE

The work described in this report was authorized under Contract No. DAAD05-91-P6564, Back-Propagation Network for Analog Signal Separation in High Noise Environments. This work was started in August 1991 and completed in January 1992.

The use of trade names or manufacturers' names in this report does not constitute an official endorsement of any commercial products. This report may not be cited for purposes of advertisement.

This report has been approved for release to the public.

Blank

CONTENTS

LIST OF FIGURES

LIST OF TABLES

BACK-PROPAGATION NETWORK FOR ANALOG SIGNAL SEPARATION IN HIGH NOISE ENVIRONMENTS

1. INTRODUCTION

Chromatography and photospectrometry are techniques commonly used to identify the composition of mixtures. These spectra are comprised of an additive combination of the individual spectrum and often the individual spectrum overlap and interfere with one another thus necessitating the need for some signal separation algorithm. Traditionally, principal components regression (PCR) is used to perform this task[1]. Furthermore, the concentration of the component in question may be so low that it is near the detection limit of the apparatus in use, thus the signal may be very noisy. Such a situation occurs in biological and chemical weapons detection because one wishes to alarm at the earliest possible time, i.e., as soon as the concentration reaches the detection limit of the warning device. Another inherent problem with biological and chemical weapons detection is that the battlefield conditions are constantly changing and therefore the background noise will also change. For these reasons it would be advantageous to have an adaptable detection system that is capable of performing in high noise environments.

Other researchers have successfully applied artificial neural networks to component separation problems using only two components and little added noise[2]. In this paper the back-propagation (BP) network is examined as a possible alternative approach to PCR and prefiltered linear regression (PLR) for separation of up to four components in a high noise environment.

2. BACKGROUND

2.1 Linear Regression

In this paper scalars will be denoted by italic lowercase letters, vectors by bold lowercase letters, matrices by bold uppercase letters, and the transpose by a superscript T. Linear regression assumes

$$\mathbf{D} = \mathbf{C}\mathbf{S}^{T} + \mathbf{E}$$

where **D**, the data matrix, is dimensioned i X j, **S**, the sensitivity matrix, is dimensioned j X k, and **C**, the matrix of concentrations, is dimensioned i X k. **E** is a matrix of response residuals. The sensitivity matrix can be estimated by

$$\mathbf{S}' = \mathbf{D}^{T}\mathbf{C}(\mathbf{C}^{T}\mathbf{C})^{-1}$$

where the relationship between **D** and **C** is known. This set will be referred to as the training set. The set **C**′ and **D**′, where **C**′ is the unknown concentrations of the data **D**′, will be referred to as the test set. The unknown concentrations **C**′ can then be estimated using

$$\mathbf{C}' = \mathbf{D}'^{T}\mathbf{S}'(\mathbf{S}'^{T}\mathbf{S}')^{-1}$$

---

1 Kowalski, B. and Seasholtz, M., "Recent Developments in Multivariate Calibration," Journal of Chemometrics Vol. 5, pp 129-145 (May 1991).

2 Long, J., Gregoriou, V., and Gemperline, P., "Spectroscopic Calibration and Quantitation Using Artificial Neural Networks," Analytical Chemistry Vol. 62, no. 17, pp 1791-1797 (Sept. 1990).

2.2 Principal Components Regression

PCR uses the same equations as linear regression except it replaces **D** and **D´** with **N** and **N´** given by

$$\mathbf{N} = (\mathbf{R}^T\mathbf{D}^T)^T$$
$$\mathbf{N}' = (\mathbf{R}^T\mathbf{D}'^T)^T$$

where **R** is the first n eigenvectors of $\mathbf{D}^T\mathbf{D}$ and will be referred to as the principal components. The number of principal components, n, is determined by the size of the relative eigenvalues, i.e., the larger the eigenvalue the more important the relative eigenvector is. **N** and **N´** are dimensioned n X i and **R** is dimensioned j X n. The step of eliminating everything other than the principal components acts to keep only the significant information in the data and is somewhat analogous to spectral filtering.

2.3 The Back-Propagation Network

The BP network described below uses the generalized delta rule learning and the layout of a typical BP network and is consistent with the equations and layout used for this study[3]. It should be noted, however, that there are many variations on the standard algorithm and detailed descriptions of these modifications are available in the literature[4, 5].

A BP network typically has an input, output, and at least one hidden layer. It usually also has a bias element which outputs to all the elements in the hidden and output layers. Normally the input and hidden layer and the hidden and output layer are fully interconnected, meaning all the processing elements (PE's) of one layer are connected to all the PE's of the other layer. All the connections between the PE's contain weights that act as gains along those paths. Each PE sums all its inputs, modifies it by some transfer function, and outputs the resulting value. The network learns by repetitiously presenting the input/output (I/O) pairs contained within the training set, forward propagating the inputs through to the output layer, and modifying the connection weights by back propagating a modified error function which is based on the result of the forward propagated input compared with the output portion of the I/O pair. The forward propagation step is accomplished by

$$x_j[s] = F\{ \Sigma_i (w_{ji}[s]\ x_i[s-1]) \}$$

where

$x_j[s]$ = output of the jth PE in layer s
$w_{ji}[s]$ = connection weight between the ith PE in layer (s-1) and the jth PE in layer s
$F\{\bullet\}$ = the PE's transfer function

---

3 Jones, W. and Hoskins, J., "Back-Propagation, A Generalized Delta Learning Rule," Byte Magazine (Oct. 1987).

4 McClelland, J. and Rumelhart, D., Explorations in Parallel Distributed Processing, The MIT Press, Cambridge, Mass. (1988).

5 Wasserman, P., Neural Computing, Theory and Practice, Van Nostrand Reinhold, New York, New York (1989).

Some of the more common transfer functions are

Sigmoid	$F\{z\} = (1.0 + e^{-gz})^{-1}$
Hyperbolic Tangent	$F\{z\} = (e^{gz} - e^{-gz}) / (e^{gz} + e^{-gz})$
Linear	$F\{z\} = z$

where g is called the gain. Once the input has been forward propagated to the output layer an error term is generated and back propagated using

$$e_j[s] = F'\{ \Sigma_i (w_{ji}[s]\ x_i[s-1])\} \Sigma_k (e_k[s+1]\ w_{kj}[s+1])$$

where

$e_j[s]$ = error term for the jth PE in layer s
$F'\{\bullet\}$ = the first derivative of $F\{\bullet\}$

and the error term for the output layer is given by

$$e_j[s_o] = F'\{\Sigma_i(w_{ji}[s_o-1]\ x_i[s_o-1])\}\ (d_j - x_j[s_o])$$

where d_j = the desired output of the jth PE given by the present I/O pair. After the error term has been calculated the given connection weight is modified by

$$\Delta w_{ji}[s] = c\ e_j[s]\ x_i[s-1] + m\ \Delta w'_{ji}[s]$$

where

$\Delta w_{ji}[s]$ = delta weight between the ith PE in layer (s-1) and the jth PE in layer s
$\Delta w'_{ji}[s]$ = previous delta weight for the given connection and layer
c = learning coefficient
m = momentum term

The above steps will be repeated until the designer feels that the network is sufficiently well trained. Usually the training pairs are presented to the network in a random fashion to prevent it from overlearning some arbitrary patterns resulting from the location of the training pairs with respect to one another. Once the network is trained it is normally tested with a different data set called the test set to better evaluate the models generality.

3. EXPERIMENTAL SETUP

3.1 Generation of the Data

The data was comprised of an additive combination of the individual spectrum. The individual spectrum were generated using

$$y_i = e^{-(x-m)\wedge 2 / sx}$$

where

m = the peak location
s = skewing factor

A plot of the individual spectrum for the four component mixture can be seen in Figure 1. The individual spectrum were represented by 45 points each and were combined into an input spectrum using

$$Y_j = \Sigma_i c_{ij} y_i \quad j = 1, 2, \ldots, n_k$$

where

Y_j = the jth input spectrum
c_{ij} = concentration of component y_i for the jth input spectrum
n_k = number of possible combinations of concentrations for the kth matrix

and the c_{ij}'s were determined by

$$\Sigma_i c_{ij} = 1 \quad j = 1, 2, \ldots, n_k$$

and all possible combinations of the incremental concentrations were generated to make the input matrix. The training set concentrations for the two, three, and four component mixtures were incremented at 10, 20, and 25 percents intervals respectively. The test set concentrations were incremented at 1, 1, and 2 percent intervals respectively. Each training matrix was the replicated 10 times to make the final size of the input training matrix 45 by $10n_k$ and each test matrix was replicated 5 times to make the final size of the input test matrix 45 by $5n_k$. For each input matrix a noise matrix consisting of uniformly distributed random variables between 0 and 0.3 was generated and added to the input matrix. Finally, to each Y_j was added a randomly chosen constant between 0 and 1. Figure 2 shows the first tenth of the resulting training set for the four component input matrix.

3.2 Filtering Methodology

No prefiltering was performed for the BP network. For PCR and PLR the random dc was removed by subtracting the average of the first and last three points. For PLR further filtering was accomplished by multiplying the spectral domain by a square function which equaled one between -p and p and was zero elsewhere. The cutoff frequency, p, for this low pass filter was optimally chosen by comparing the filter output with the input matrix prior to the addition of the noise.

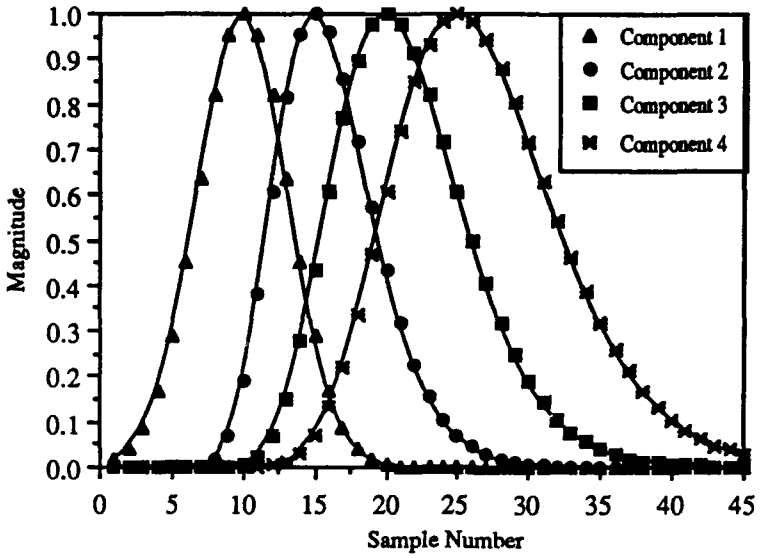
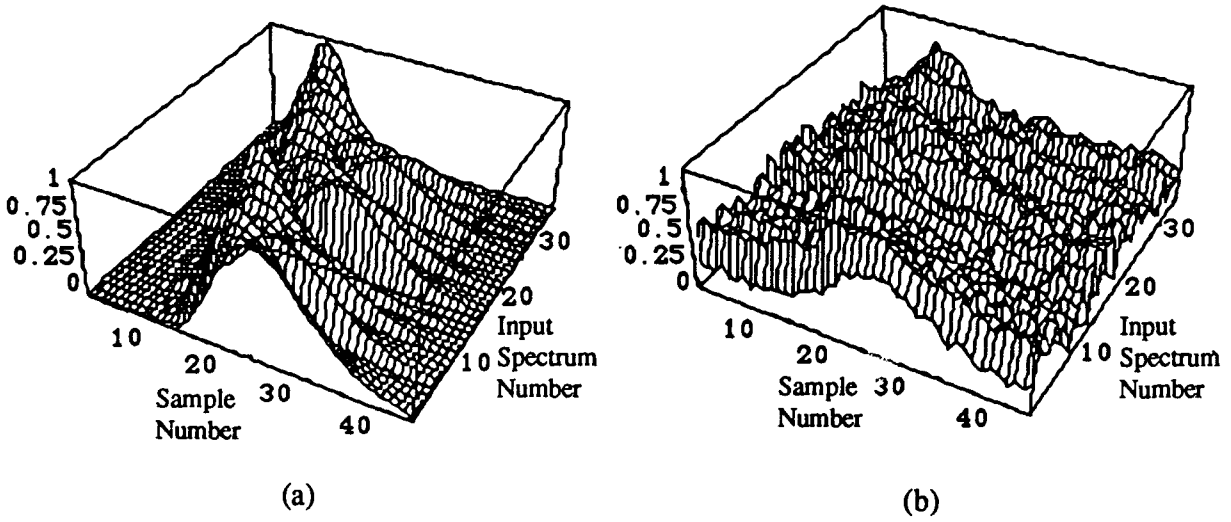


Figure 1. The individual spectrum plotted together.



(a) (b)

Figure 2. (a) Four component matrix without the noise added.
(b) Four component matrix with the noise added.

3.3 Principal Components Regression

The number of principal components was determined to be q where q equals the number of components in the mixture. This was determined by plotting the eigenvalues as is shown in Figure 3 for the four component mixture. The eigenvalues dropped off very rapidly and remained fairly constant after the qth eigenvalue.

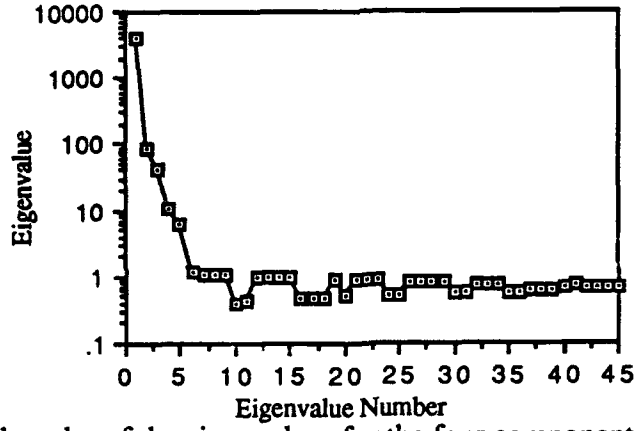


Figure 3. A log plot of the eigenvalues for the four component input matrix.

3.4 Back-Propagation Network Setup

The BP network had a bias PE, 45 input PE's, q hidden PE's, and q-1 output PE's where q equals the number of components in the mixture. The number of hidden PE's was determined by starting at a large number and progressively removing inactive PE's. Both q and q-1 output PE's were tried initially and the results for q-1 were slightly better thus the remaining experiments were performed with this number. The hidden PE's had sigmoid transfer functions with gain set to one and the input and output PE's had linear transfer functions. Sigmoid transfer functions on the output PE's tended to warp the output and resulted in greater error. The network was trained for the two, three, and four component mixtures with 50,000, 60,000, and 70,000 iterations respectively.

4. RESULTS AND DISCUSSION

4.1 Comparison Between Methodologies

The results are given in terms of the error averaged over the entire set of training pairs and components and is shown in Table I. In all cases the BP network outperformed PCR and PLR. The error for PCR and PLR was between 30 and 65 percent greater than that for the BP network. The difference between PCR and PLR as compared with the BP network decreased as the number of components increased. This occurred because the error for the BP network as a function of the number of components had a greater slope than it did for the other methods. To understand this one must first explore the two primary sources of error that vary with the number of components. The first source of error is caused by the degree of signal distortion due to the heavily overlapped nature of the spectra in question. This error increases equally for all three methods and does not account for the apparent discrepancy. The other primary source of error that increases as a function of the number of components is exclusive to the BP network and is caused by the fact that an increase in the number of components results in a linearly related increase in the number of weights. This makes the weight space more complex and thus the gradient search like minimization algorithm of the BP network will have greater difficulty finding the global minima and can easily get stuck in local minima. This does not occur in PCR because the eigenvector/eigenvalue search is performed in 45 by 45 space for all cases.

TABLE 1 - RESULTS OF THE COMPARISONS BETWEEN METHODS

Method	Average Percent Error		
	Two Component	Three Component	Four Component
BP Network	2.365	3.530	4.448
PCR	3.906	5.104	5.886
PLR	3.639	4.810	5.744

4.2 Low Noise High Noise Comparison for the Back-Propagation Network

It would be a useful characteristic if one could train the BP network with a high noise worst case scenario and recall with input test sets that varied from little noise up to the worst case. In order to determine whether the BP network could manage this task I proceeded to recall the network which had been previously trained on the thirty percent noise data with a new data set containing only ten percent noise. The results are shown in Table II. As expected the average error decreased significantly with the new data set.

TABLE 2 - RESULTS OF THE COMPARISON BETWEEN THE LOW AND HIGH NOISE TEST SETS FOR THE BACK-PROPAGATION NETWORK

Test Set Error	Average Percent Error		
	Two Component	Three Component	Four Component
Ten Percent	1.503	1.503	1.906
Thirty Percent	2.365	3.530	4.448

5. CONCLUSIONS

Using BP networks for signal separation seems to have several advantages over classical linear regression based techniques. The apparent ability of the network to generalize would seem to indicate that it is possible to initially train the network with the worst case scenario, thus allowing it to generalize about the information content, and then recall with data that can vary anywhere from perfect up to the worst case. The network is also not restricted to purely linear relationships in the data. In this paper only linear relationships existed but if data containing some nonlinearities was used the BP network should fare even better in comparison with the linear techniques. Finally, using a neural network type approach should allow the network to constantly update itself as the background noise varies, thus providing some degree of adaptability.